Online Inventory Management with Application to Energy Procurement in Data Centers

Lin Yang*
Chinese University of Hong Kong
yl015@ie.cuhk.edu.hk

Mohammad H. Hajiesmaili*
UMass Amherst
hajiesmaili@cs.umass.edu

Ramesh Sitaraman
UMass Amherst and Akamai
ramesh@cs.umass.edu

Enrique Mallada
Johns Hopkins University
mallada@jhu.edu

Wing S. Wong
Chinese University of Hong Kong
wswong@ie.cuhk.edu.hk

Adam Wierman
California Institute of Technology
adamw@caltech.edu

ABSTRACT

Motivated by the application of energy storage management in electricity markets, this paper considers the problem of online linear programming with inventory management constraints. Specifically, a decision maker should satisfy some units of an asset as her demand, either form a market with time-varying price or from her own inventory. The decision maker is presented a price in slot-by-slot manner, and must immediately decide the purchased amount with the current price to cover the demand or to store in inventory for covering the future demand. The inventory has a limited capacity and its critical role is to buy and store assets at low price and use the stored assets to cover the demand at high price. The ultimate goal of the decision maker is to cover the demands while minimizing the cost of buying assets from the market.

We propose `BatMan`, an online algorithm for simple inventory models, and `BatManRate`, an extended version for the case with rate constraints. Both `BatMan` and `BatManRate` achieve optimal competitive ratios, meaning that no other online algorithm can achieve a better theoretical guarantee. To illustrate the results, we use the proposed algorithms to design and evaluate energy procurement and storage management strategies for data centers with a portfolio of energy sources including the electric grid, local renewable generation, and energy storage systems.

CCS CONCEPTS

• **Theory of computation** → **Online algorithms**; • **Hardware** → **Energy generation and storage**; **Enterprise level and data centers power issues**.

ACM Reference Format:
Lin Yang, Mohammad H. Hajiesmaili, Ramesh Sitaraman, Enrique Mallada, Wing S. Wong, and Adam Wierman. 2019. Online Inventory Management with Application to Energy Procurement in Data Centers. In *Proceedings of Submitted*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Both authors contributed equally to this research.

Submitted, 2019,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online optimization and decision making under uncertainty is a fundamental topic that has been studied using a wide range of theoretical tools and in a broad set of applications. On the theoretical side, it has been approached from the perspective of competitive algorithms with the competitive ratio as the performance metric [1], online learning with the regret as the performance metrics [2, 3], and reinforcement learning with different modeling techniques such as Markov decision process [4]. On the application side, recent scenarios where theoretical results have had an impact for real-world design include data center optimization [5–9], energy systems [10–13], cloud management [14–16], computer and communication networks [17–20], and beyond.

Motivated by storage management problem for data centers procuring energy from the electricity market, this paper studies a generalization of the classical online optimization formulation: *online linear programming with inventory management* (OLIM). In this problem, in each slot, a decision maker should satisfy $d(t)$ units of an asset, e.g., energy demand, as her demand, either from a market with time-varying price or from her own inventory, e.g., energy storage system. In slot t, the decision maker is presented a price $p(t)$, and must decide $x(t)$ as the procurement amount with the current price to cover the demand $d(t)$ or additionally to store in inventory for covering the future demand. The capacity of inventory is B units, and its critical role is to buy and store assets at low price and use those for covering the demand at high price. The goal is to cover the demands while minimizing the cost of buying from the market. A formal statement of the problem is presented in § 2.

OLIM captures a variety of timely applications in different domains, e.g., booking hotels or flight tickets in advance to for high season by travel agencies or charging energy storage in data centers to use in high price periods. The application that motivates our interest in studying OLIM is designing energy procurement and storage management strategies for large-scale electricity customers such as data centers, university campuses, or enterprise headquarters. Usually, these customers can satisfy their energy demand from a portfolio of sources, including the grid, local renewable sources, and on-site energy storage systems. Notable examples are thermal energy storage in Google data center in Taiwan [21], Tesla batteries to power Amazon data center in California [22], Google data center with on-site renewable sources in Belgium [23], and large-scale batteries in Apple Park's microgrid [24]. The electricity pricing for large customers is moving toward real-time pricing and the price changes dynamically over time [10, 25, 26]. The addition of on-site

storage systems presents a great opportunity for shifting energy usage over time to reduce the energy cost by purchasing the energy at low price periods. The severe uncertainty in energy demand and price, however, make designing optimal energy procurement strategies a challenging task and emphasizes the need for *online solution design*. In §2, we present the detailed energy procurement scenario and the challenges due to uncertainty in the problem.

Note that OLIM is a generalization of several classic online algorithmic problems. The first category is online search for the optimum with sequential arrival of online price, known as online search or conversion problems [27]. Notable examples are the time series search and one-way trading problems [28], the multiple-choice secretary problem [29], the k-search problem [30], and online linear programs with covering constraints [31]. Different from these problems, in OLIM, in addition to the uncertainty in the market pricing, we have another uncertainty due to online arrival of the demand. In other words, OLIM comes with two sets of uncertain input parameters, which allow the adversary to have more options in constructing the worst-case input. In terms of constraints, OLIM includes inventory management constraints that couples the covering constraints over time. More details are given in §2.

Summary of Contributions. In this paper, we develop a online algorithms for OLIM and show that the algorithm achieves the minimal competitive ratio achievable by an online algorithm. More specifically, we propose `BatMan`[1] (§3.1) and `BatManRate` (§3.3). `BatMan` is a simpler algorithm that works for the inventory that have no rate constraints, i.e., no limit on input and output rate to/from the inventory at any slot. `BatManRate` works in a more general context where the input and output (a.k.a., charge and discharge, in the application context) rates are bounded.

The high-level intuition behind the design of `BatMan` is to store assets at cheap and use the stored asset once the price is expensive. However, the dynamic pricing and dynamic demand make this decision making challenging. The main ideas of `BatMan` are: (i) *adaptive reservation based on storage*[2] *utilization* that tackles the challenges due to price dynamics; and (ii) *construction of virtual storages* that tackle the challenges due to demand dynamics. The idea of using adaptive pricing function is adapted from the online algorithms for k-search problem in [30].

The main novelty in the algorithm design is introducing the novel notion of virtual storages to tackle the additional demand uncertainty. In particular, given some back-up assets in inventory, one can see satisfying the demand in each slot as as the buying (minimization) version of an optimal search problem with the current demand as the target amount. However, dynamic arrival of demands makes these online search problems coupled over time, and exacerbates the competitive analysis of the algorithms.

Our main technical results provide an analysis of the competitive ratios of `BatMan` and `BatManRate`. These results are summarized in Theorems 1.1 and 1.2.

Theorem 1.1. *With the following reservation function*

$$G_{B_i}(p) = \alpha B_i \ln\left[\left(1-\frac{p}{p_{\max}}\right)\frac{\alpha}{\alpha-1}\right],\ p \in \left[p_{\min}, \frac{p_{\max}}{\alpha}\right], \quad (1)$$

`BatMan` *achieves the optimal competitive ratio of* α *defined as*

$$\alpha = \left(W\left(-\frac{\theta-1}{\theta\exp(1)}\right)+1\right)^{-1}. \quad (2)$$

In Equation (2), θ is the price fluctuation ratio, and $W(.)$ is Lambert-W function, defined as the inverse of $f(z) = z\exp(z)$.

Interestingly, the above competitive ratio for `BatMan` *is exactly the same as the optimal competitive ratio for k-min search problem (see [30, Theorem 2]), when $k \to \infty$.* However, OLIM involves additional uncertainty on demand as compared to [30] and additional inventory management constraint. The additional demand uncertainty enlarges the design space of the adversary and complicates the competitive analysis. To obtain the performance bounds of online and offline algorithms, we introduce several novel techniques and notions, such as definition of reservation and idle periods.

Theorem 1.2. `BatManRate` *achieves the optimal competitive ratio of* α *as in Equation* (2).

`BatManRate` extends `BatMan` to the case with rate constraints. Two significant changes in algorithm design are adaptive determination of the capacity of virtual storages to reflect the output rate, and adaptive setting of reservation price to reflect the input constraints. While the general logic for the analysis of `BatManRate` is similar to that of `BatMan`, it comes with a significant result on showing that in worst-case, the output rate constraint is not active.

In addition to providing theoretical analysis of `BatMan`, in §4, we also empirically evaluate the performance of the proposed algorithms using real-world data traces in the data center energy scenario. More specifically, evaluate our algorithms using extensive data traces of electricity prices from several electricity markets (CAISO [32], NYISO [33], ERCOT [34], DE Market [35]), energy demands from multiple data centers of Akamai's CDN [36], and renewable production values from solar [37] and wind installations [38, 39]. In a broad set of representative scenarios that include different seasons and locations, `BatMan` achieves a cost reduction of 15% in comparison with using no energy storage at all, establishing the value of batteries for energy procurement. Further, `BatMan` achieves an energy procurement cost that is within 21–23% of the theoretically smallest achievable cost in an offline setting. In addition, `BatMan` outperforms state-of-the-art algorithms for energy procurement by 7.5–10%. Finally, `BatManRate` outperforms all the alternatives in our experiments.

2 PROBLEM FORMULATION

We present the system model and formulate the problem. The interpretation of the model parameters is illustrated by the application of storage-assisted energy procurement in electricity market.

We assume that the time-slotted model in which the time horizon is divided into T slots, indexed by t, each with fixed length, e.g., 5 minutes in California ISO (CAISO) and New York ISO (NYISO) [40]. We consider the following scenario. At each slot, a demand $d(t)$ arrive online that must be satisfied from either the market with the real-time price $p(t)$ or from the local inventory, i.e., energy storage system. The decision maker can purchase more from the market and store in the inventory to satisfy the future demand. The ultimate goal is to design an algorithm to determine the value of $x(t)$, as the procurement amount in each slot, such that the procurement cost is

[1] `BatMan` is short for Battery Management, inspired from our application of interest in optimizing energy procurement by *battery (energy storage) management*.
[2] Throughout the paper, *inventory* and *storage* are used interchangeably.

minimized over a time horizon, and the demand is satisfied. In the following, we introduce the inventory management constraints.

Inventory Management Constraints. Let B, ρ_c, and ρ_d be the capacity, the maximum input rate, and the maximum output rate of the inventory. Let $b(t) \in [0, B]$ be the inventory level at the end of slot t that represents the amount of assets that are already in the inventory. The evolution of inventory level is given by

$$b(t) = b(t-1) + x(t) - d(t), \tag{3}$$

which states that the amount of assets in the inventory at the end of each round is equal to the previous existing amount $(t-1)$ and the current procurement amount $x(t)$ subtracted by the demand $d(t)$. Moreover, we have two constraints on the value of $x(t)$:

$$x(t) \geq d(t) - \min\left\{\rho_d, b(t-1)\right\}, \tag{4}$$

which captures the maximum output rate from the inventory and ensures covering the demand, and

$$x(t) \leq d(t) + \min\left\{\rho_c, B - b(t-1)\right\}, \tag{5}$$

which captures the input rate constraint to the inventory. Finally, we have the following inventory capacity constraint

$$0 \leq b_{\mathrm{d}}(t) \leq \min\{\rho_d, b(t-1)\},$$

In our example of energy storage, the rate constraints ρ_c and ρ_d are the charge and discharge rate of the energy storage systems. Several examples of actual values of ρ_c and ρ_d for different energy storage technologies, e.g., lead-acid and lithium-ion batteries, and compressed air energy storage are provided in §4.2.

Problem Formulation. We can now summarize the full formulation of online optimization with inventory management (OLIM). If the demands and market prices are known for the entire time horizon in advance, the *offline* version of OLIM can be formulated a linear program as follows.

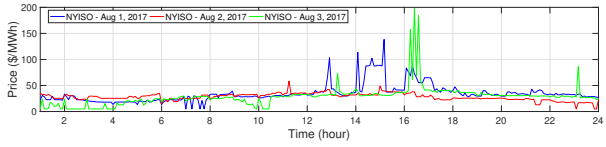
$$\begin{aligned}
\text{OLIM}: \quad \min \quad & \sum_{t\in\mathcal{T}} p(t)x(t) \\
\text{s.t.}: \quad & \forall t \in \mathcal{T}: \\
& x(t) \geq d(t) - \min\left\{\rho_d, b(t-1)\right\}, && (6)\\
& x(t) \leq d(t) + \min\left\{\rho_c, B - b(t-1)\right\}, && (7)\\
& b(t) = b(t-1) + x(t) - d(t), && (8)\\
& 0 \leq b(t) \leq B, && (9)\\
\text{vars.}: \quad & x(t) \in \mathbb{R}^+.
\end{aligned}$$

The objective is to minimize the procurement cost from the market. Constraints (6)-(7) ensure covering the demand and rate limits. Constraint (8) dictates the evolution of the inventory, and (9) enforce the capacity of the inventory. Since OLIM is a linear program it can be solved efficiently in an offline manner.

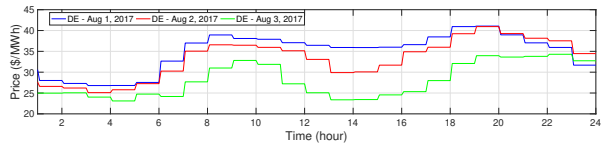
In this work, we are interested in developing *online algorithms* for OLIM that make decisions at each time t, knowing the past and current prices and demands, but not knowing those same inputs for the future. The algorithmic challenge is to procure assets from the market and store in the storage in the current time, without knowing if such decisions will work out favorably in the future. The classical approach for evaluating online algorithms is competitive analysis, where the goal is to design algorithms with the smallest *competitive ratio*, that is, the cost ratio between the online algorithm and an *offline* optimal algorithm that has access to *complete* input sequence. In our work, we devise an online algorithm that has provably the best competitive ratio for OLIM. In the design of algorithms, we assume that the values of $p_{\max}$ and $p_{\min}$, as the maximum and minimum prices, are known a priori. This assumption is reasonable since by the historical data, these values could be predicted. Further, related problems makes the similar assumptions [27, 28, 30]. Let $\theta = p_{\max}/p_{\min}$ as the price fluctuation ratio. Our analysis characterizes the performance as a function of θ.

The Case Study. Our motivation for studying OLIM comes from energy storage management in electricity markets. In such scenarios the demand and price values are highly uncertain and unpredictable. To further motivate the online algorithm design, in the following, we demonstrate the uncertainty of these values using real data-traces from electricity markets and Akamai data centers.

1) The electricity pricing for large customers like data centers is moving toward real-time pricing and the price changes dynamically over time [41–45]. Two examples of real-time energy prices in NYISO and DE Market are demonstrated in Figure 1. By comparing the price dynamics in two different electricity market, we can see totally different patterns. While the prices in NYISO highly fluctuate without any regular pattern, in German Electricity Market, we observe regular daily patterns with low price fluctuations.



(a) NYISO



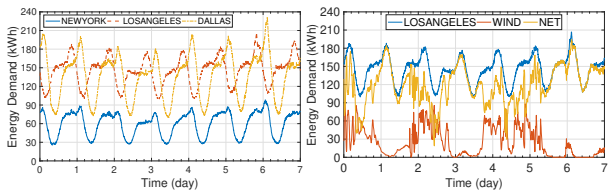
(b) German Electricity Market

Figure 1: The energy price dynamics in NYISO and German (DE) Market in three consecutive days in August 2017

2) The data center energy demand is highly unpredictable because user demand for Internet services is extremely variable. Further, the sophisticated optimization algorithms used to improve the energy efficiency of data center's internal operations [46] can further increase the unpredictable variability of the energy demand.

In addition, recently, several data centers are equipped with on-site renewable sources, e.g., Google data center in Belgium [23]. The energy production level of renewable sources is uncertain and intermittent (exhibits high fluctuations) [47]. For instance, energy production from solar panels can change in a matter of minutes due to cloud cover moving in to obscure the sun. This may lead to an increased uncertainty in net energy demand of data center.

To represent typical data center energy demand with and without renewable sources, we collected the energy consumption of Akamai's server clusters in different cities, some of which are depicted in Figure 2(a) (details in Appendix E). The server clusters are



(a) United States (b) Net energy demand with wind

Figure 2: Data center energy demand in different locations

part of Akamai's global CDN [36] that serves about a quarter of the Web traffic worldwide. We observe that the total energy consumption in different data centers has a relative regular (daily) pattern. However, by injecting renewable generation, the net demand, as depicted in Figure 2(b), exhibits a large degree of variability.

Related Algorithmic Problems. Note that OLIM is related to several algorithmic problem including one-way trading [28], secretary problem [29], k-search problem [30], and online linear programming with covering constraints [31]. It is related since in all above problems and OLIM, the goal is to search for the optimum. In contrast, OLIM is unique since it is the first that tackles inventory management constraints.

Perhaps the most closely related problem to OLIM is the minimization variant of k-search, known as k-min search problem [30]. In this problem, a player wants to buy $k \geq 1$ units of an asset with the goal of minimizing the cost. At any slot $t = \{1, \ldots, T\}$, the player is presented a price $p(t)$, and must immediately decide whether or not to buy *some integer units* of the asset given $p(t)$. By setting $d(t) = 0, t \in \{1, \ldots, T-1\}$ and $d(T) = B$ and allowing *fractional purchase*, OLIM degenerates to the k-min search problem. Hence, OLIM is an extension of continuous version of k-min search problem, or equivalently the minimization version of one-way trading problem since one-way trading problem can be viewed as the k-max search problem with $k \to \infty$.

The other category is online linear programs with covering constraints [31]. As compared to [31], OLIM has a more specific category of covering constraints, however, the inventory management constraint in OLIM couples the covering constraints across different time slots, which results in more challenging problem than basic online covering linear programs.

3 OPTIMAL ONLINE ALGORITHMS

In §3.1, we propose BatMan, an online algorithm for a basic version of OLIM without rate constraints. Then in §3.2, we prove that BatMan achieves the optimal competitive ratio. In §3.3 and based on the insights from design of BatMan, we propose BatManRate that tackles the general OLIM problem with rate constraints, and prove that it achieves the optimal competitive ratio.

3.1 BatMan: An Online Algorithm

Suppose that the inventory had only capacity constraints, but no rate constraints, i.e., $\rho_c = \rho_d = B$. In OLIM, one can rewrite constraint (6) as $x(t) \geq d(t) - b(t-1)$, and remove constraint (7). In the next, we propose BatMan that finds a solution to OLIM without rate constraints in online manner.

3.1.1 The Design of BatMan. The high-level idea of designing BatMan is to store the asset when the market price is cheap and use the stored asset when the price is expensive. However, the dynamic pricing and dynamic demand make this decision making challenging. The main ideas of BatMan are: (i) *adaptive reservation based on storage utilization* that tackles the challenges due to price dynamics; and (ii) *construction of virtual storages* that tackle the challenges due to demand dynamics.

Adaptive Reservation Price. BatMan deals with the price dynamics by defining the notion of *reservation price*. Having a properly constructed reservation price, BatMan stores the asset if the current price is cheaper than the reservation price; otherwise, it releases the asset from the storage. BatMan adaptively determines the reservation price based on the available storage space. Intuitively, once the storage level is low, it is more eager to store asset, hence, it accepts higher prices. On the other hand, at high storage utilization, it stores the asset if the price is low. This is different from fixed reservation design introduced in [45] that determines the reservation prices without considering the current storage utilization.

Constructing Virtual Storages. BatMan deals with the demand dynamics by defining the notion of virtual storages. BatMan views the demand in each time slot as an asset that must be purchased from the market with some degree of freedom obtained by shifting it using the storage. To utilize this opportunity, BatMan constructs several virtual storages to record the satisfied amount of the demand from the market. Specifically, in each slot with $d(t) > 0$, BatMan initiates a virtual storage whose capacity is equal to the demand, and its initial level is empty. BatMan also renews the virtual storage units once the actual storage becomes empty.

The Details of BatMan. By summarizing the pseudocode of BatMan in Algorithm 1, we proceed to discuss the details. In all algorithms and analysis, we assume that the initial storage level is zero, i.e., $b(1) = 0$. For notational convenience, we represent the physical storage as the first virtual storage and define $B_1 = B$ as its capacity (Line 2) and v is the current number of virtual storages given positive demand that evolves over time (Line 6 indicates creating a new virtual storage for the current slot and Line 13 renews the virtual storages). Let B_i be the capacity of i-th virtual storage and $B_v = d(t)$, i.e., we set the value of new virtual storage to the current demand. Let ξ_i be the reservation price associated to the virtual storage i with initial value of $p_{\max}/\alpha$, where $\alpha > 0$ is a parameter that will be carefully chosen based on the competitive analysis.

The cornerstone of BatMan is in Line 10 where the procurement amount for each virtual storage, i.e., $x_i(t)$, is set. BatMan defines $G_{B_i}(\xi_i)$ as the reservation function to determine the amount of asset to be stored in the i-th virtual storage in each slot. This function represents the target amount of stored assets in i-th virtual storage when the reservation price is ξ_i. In slot t, BatMan stores additional amount of $G_{B_i}(p(t)) - G_{B_i}(\xi_i)$ into virtual storage i if the current price, $p(t)$, is less than ξ_i; otherwise, it stores nothing. Both situations can be stated in the following compact form

$$x_i(t) = \left[G_{B_i}(p(t)) - G_{B_i}(\xi_i)\right]^+. \tag{10}$$

In this way, BatMan logically allocates $x_i(t)$ units of asset to storage i (Line 10), and the reservation price will be updated to $\min\{\xi_i, p(t)\}$

Algorithm 1 The BatMan algorithm for each $t \in \mathcal{T}$

1: // Initialization: at $t = 1$
2: $B_1 \leftarrow B$; // the capacity of physical storage
3: $\upsilon \leftarrow 1$; // the number of virtual and physical storages
4: $\xi_1 \leftarrow [p_{\max}/\alpha]$; // reservation price of physical storage
// The main algorithm for t
5: **if** $d(t) > 0$ **then**
6: $\quad \upsilon \leftarrow \upsilon + 1$
7: $\quad B_\upsilon \leftarrow d(t)$
8: $\quad \xi_\upsilon \leftarrow p_{\max}/\alpha$
9: **end if**
10: $x_i(t) \leftarrow [G_{B_i}(p(t)) - G_{B_i}(\xi_i)]^+, \quad \forall 1 \le i \le \upsilon$
11: $\xi_i \leftarrow \min\{p(t), \xi_i\}, \quad \forall 1 \le i \le \upsilon$
12: $x(t) \leftarrow \max\left\{\sum_{i=1}^{\upsilon} x_i(t), [d(t) - b(t-1)]^+\right\}$
13: $b(t) \leftarrow b(t-1) + x(t) - d(t)$
// Renew virtual storages
14: **if** $b(t) = 0$ **then** $\upsilon \leftarrow 1; \xi_1 \leftarrow p_{\max}/\alpha$;

(Line 11). In other words, ξ_i records the minimum seen market price during the lifetime of i-th virtual storage.

The main contribution of BatMan is the design function $G_{B_i}(p)$ such that it achieves the optimal competitive ratio. To accomplish this, we choose the following function:

$$G_{B_i}(p) = \alpha B_i \ln\left[\left(1 - \frac{p}{p_{\max}}\right)\frac{\alpha}{\alpha - 1}\right], \; p \in \left[p_{\min}, \frac{p_{\max}}{\alpha}\right], \quad (11)$$

where

$$\alpha = \left(W\left(-\frac{\theta - 1}{\theta \exp(1)}\right) + 1\right)^{-1}, \quad (12)$$

and W denotes *Lembert-W function* defined as inverse of $f(z) = z\exp(z)$, and $\theta = p_{\max}/p_{\min}$ is the price fluctuation ratio. Figure 3(a) depicts function $G_{B_i}(p) \in [0, B_i]$, and $p \in [p_{\min}, p_{\max}/\alpha]$ as a decreasing function. It also demonstrates how to determine the reservation amount for virtual storage i. Further, when the price p is larger than and equal to $p_{\max}/\alpha$, $G_{B_i}(p) = 0$. When the price is equal to $p_{\min}$, the reservation amount is equal to $\alpha B_i \ln\left[\left(1 - \frac{1}{\theta}\right)\frac{\alpha}{\alpha-1}\right]$. By substituting the value of α from Equation (12), we have $G_{B_i}(p_{\min}) = B_i$, which means when the price is minimum, BatMan stores the full capacity of the storage.

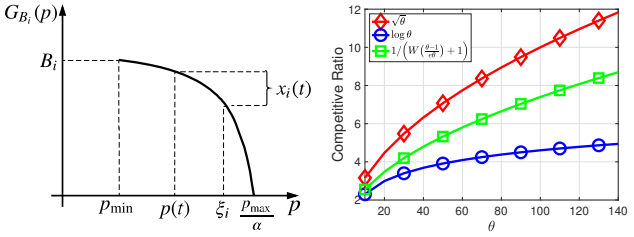
The last step is to determine the aggregate procurement quantity $x(t)$ (Line 12). To satisfy the demand constraint, i.e., $x(t) \ge d(t) - b(t-1)$, we calculate $x(t)$ as follows

$$x(t) = \max\left\{\sum_{i=1}^{\upsilon} x_i(t), d(t) - b(t-1)\right\}. \quad (13)$$

When $\sum_{i=1}^{\upsilon} x_i(t) < d(t) - b(t)$, BatMan discharges the physical storage completely to meet the demand, hence, we have $b(t) = 0$. In this case, we renew all virtual storages to the initial state of having only the physical storage (Line 13). The intuition is that with fully discharging the physical storage, we exhausted the capability of shifting the demand using the storage, hence we renew the process.

Theorem 3.1. BatMan *generates a feasible solution to* OLIM.

The proof for covering the demand is straightforward because of Equation (13). The range of function $G_{B_i}(p)$ along with the renewal process in Line 13 guarantees the respecting the capacity constraint of physical storage. The proof is formally given in Appendix A.



(a) An illustration of function $G_{B_i}(p)$ and determining $x_i(t)$ as the procurement amount for virtual storage i

(b) An illustration of competitive ratio in Equation (2) as a function of θ in comparison with $\sqrt{\theta}$ [45] and $\log\theta$.

Figure 3: Function $G_{B_i}(p)$ and growth of competitive ratio

3.2 Competitive Analysis of BatMan

This section proves the result in Theorem 1.1. Before the formal proof, we state the following remark about the competitive ratio of BatMan. In Figure 3(b), we depict the value of the optimal competitive ratio α as a function of price fluctuation ratio θ, as compared to a sub-optimal competitive ratio of $\sqrt{\theta}$ achieved in [45] in a slightly different setting, and $\log\theta$ as a baseline, and it shows that its growth is less than the $\sqrt{\theta}$. However, series expansion of Lambert-W function shows that α as indicated in (2) is in order of $\Theta(\sqrt{\theta})$. This is in contrast to the best competitive ratio for equivalent maximization problems, e.g., k-max search that achieves competitive ratio of like $\log\theta$ [28, 48, 49]. This shows that minimization and maximization version of search problems behave differently in terms of best possible competitive algorithms.

In what follows, we prove the result in Theorem 1.1. First, we give the preliminaries (§3.2.1). Second, we characterize an upper bound on the cost of BatMan (Lemma 3.2). Third, a lower bound on the offline optimum is obtained (Lemma 3.3). Forth, we prove the competitive ratio by comparing these two values. Finally, we prove the optimality of the competitive ratio in §3.2.3.

3.2.1 Definitions and Preliminaries. First, to be consistent to the notations in this paper, ideally we must denote all the inputs and variables with index t. For notation brevity, however, we slightly abuse the notations by dropping index t in the analysis.

Definition 1. *Define $\omega \in \Omega$ as an input instance to* OLIM *including the price and demand, i.e.,*

$$\omega \stackrel{def}{=} [\omega(t) = \langle p(t), d(t)\rangle]_{t\in\mathcal{T}}.$$

Moreover, $\text{cost}_\omega(\text{BatMan})$ *is the cost of* BatMan *under instance ω and* $\text{cost}_\omega(\text{OPT})$ *is the offline optimal cost under ω. We drop subscript ω from the costs when we are not focusing on a particular ω.*

Definition 2. BatMan *is α-competitive, if for any $\omega \in \Omega$*

$$\text{cost}_\omega(\text{BatMan}) \le \alpha \cdot \text{cost}_\omega(\text{OPT}) + \text{cons}, \quad (14)$$

where $\text{cons} \ge 0$ *is a constant number.*

In proofs, we set the value of "cons" to capture the special input scenarios to OLIM.

Definition 3. *Reservation and idle periods. The time horizon T can be divided into two type of periods: the reservation period, which contains the interval between beginning to charge and fully discharging of the storage; and the idle period, which corresponds to the interval that lies between two adjacent reservation periods.*

The following is the list of additional notations that we use in the analysis. Consider an instance in which the executing of BatMan results in n reservation periods totally. During the i-th reservation period, $i \le n$, we assume there are totally $\hat{v}_i$ virtual storage units created. Let $\hat{\xi}_{i,j}$ be the final reservation price of the j-th virtual storage during the i-th reservation period. Let $B_{i,j}$ be the capacity of the j-th storage and $\hat{b}_{i,j}$ be the final storage level correspondingly. Obviously, we have $\hat{b}_{i,j} = G_{B_{i,j}}(\hat{\xi}_{i,j})$. Let D be the total demand during the time horizon, i.e., $D = \sum_{t\in\mathcal{T}} d(t)$, and $\hat{b}$ be the final storage level of the physical storage, i.e., $\hat{b} = b(T)$. In addition, let $F_i(\beta)$ be the minimum cost of purchasing β units of asset during i-th reservation period, and $\tilde{p}$ be the minimum price during the idle periods. Finally, the inverse of function G_{B_i} is defined as

$$G^{-1}_{B_i}(b) = p_{\max}\left[1-\left(1-\frac{1}{\alpha}\right)\exp\left(\frac{b}{\alpha B_i}\right)\right],\ b\in[0,B_i]. \tag{15}$$

3.2.2 The Proof of Theorem 1.1. The following lemma characterizes an upper bound on the cost of BatMan, during the time horizon. The proof of all lemmas in this section are given in Appendix B.

Lemma 3.2. cost(BatMan) *is upper bounded by*

$$\text{cost(BatMan)} \le \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db + \left(D-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}+\hat{b}\right)p_{\max}.$$

The next lemma yields a lower bound on the cost of offline optimal solution.

Lemma 3.3. cost(OPT) *is lower bounded by*

$$\text{cost(OPT)} \ge \sum_{i=1}^{n} F_i(\beta_i) + \left(D-\sum_{i=1}^{n}\beta_i\right)\tilde{p}.$$

Now, we proceed to prove the competitive ratio. First, we consider the simple case, where $D = 0$. In this trivial case, cost(OPT) $= 0$, that of BatMan is at most $Bp_{\max}$. Obviously, BatMan is α-competitive since it satisfies the definition of competitive ratio in Equation (14) by setting cons $= Bp_{\max}$.

Second, we focus on a realistic case, in which $D > 0$. If the minimum price during the time horizon is larger than or equal to $p_{\max}/\alpha$, the cost of BatMan is at most $p_{\max}(B + D)$ and that of OPT is lower bounded by $\frac{p_{\max}}{\alpha}D$. It is easy to see that BatMan is α-competitive according to the definition. We only consider the case where the minimum price during the time horizon is less than $p_{\max}/\alpha$, and obviously, the minimum price occurs in the reservation period. We have $\sum_{i=1}^{n} F_i(\beta_i) > 0$. Using the results in lemmas 3.2 and 3.3 we have

$$\begin{aligned}
&\frac{\text{cost(BatMan)} - \hat{b}p_{\max}}{\text{cost(OPT)}}\\
\le&\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db + \left(D-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)\cdot p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)+\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot\tilde{p}}\\
=&\frac{Q+\left(\sum_{i=1}^{n}\beta_i-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}+\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)+\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot\tilde{p}}\\
\le&\max\left\{\frac{Q+\left(\sum_{i=1}^{n}\beta_i-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)},\frac{\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot p_{\max}}{\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot\tilde{p}}\right\}\\
\le&\max\left\{\frac{Q+\left(\sum_{i=1}^{n}\beta_i-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)\cdot p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)},\alpha\right\}.
\end{aligned} \tag{16}$$

where

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db.$$

In Equation (16), the first inequality is by Lemma 3.2 and 3.3, the second inequality is by $D-\sum_{i=1}^{n}\beta_i \ge 0$ and the third inequality is by $p_{\max}/\tilde{p} \le \alpha$.

The following two lemmas provide an upper bound for Equation (16), as a main step to prove the competitive ratio.

Lemma 3.4. *Given $G^{-1}_{B_{i,j}}(\hat{b}_{i,j}), i \in \{0,1,\ldots,n\}$ in Equation* (15)*, we have*

$$\frac{\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db + (B_{i,j}-\hat{b}_{i,j})p_{\max}}{\hat{\xi}_{i,j}B_{i,j}} = \alpha,\ \forall\ \hat{b}_{i,j}\in[0,B_{i,j}]. \tag{17}$$

Proof. By substituting Equation (15), we first calculate the second term in numerator of Equation (17) as follows

$$\begin{aligned}
&\int_{b=0}^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db\\
=&p_{\max}\left[b-\left(1-\frac{1}{\alpha}\right)\exp\left(\frac{b}{\alpha B_{i,j}}\right)\alpha B_{i,j}\right]\Bigg|_{b=0}^{\hat{b}_{i,j}}\\
=&p_{\max}\left[\hat{b}_{i,j}-\left(1-\frac{1}{\alpha}\right)\exp\left(\frac{\hat{b}_{i,j}}{\alpha B_{i,j}}\right)\alpha B_{i,j}\right]\\
&+p_{\max}\left(1-\frac{1}{\alpha}\right)\alpha B_{i,j}.
\end{aligned}$$

Then, we calculate the numerator

$$\begin{aligned}
&(B_{i,j}-\hat{b}_{i,j})p_{\max}+\int_{b=0}^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db\\
=\;& \alpha B_{i,j}\left(p_{\max}\left[1-\left(1-\frac{1}{\alpha}\right)\exp\left(\frac{\hat{b}_{i,j}}{\alpha B_{i,j}}\right)\right]\right)\\
=\;& \alpha B_{i,j} G^{-1}_{B_{i,j}}(\hat{b}_{i,j}). \qquad (18)
\end{aligned}$$

Substituting (18) into (17) completes the proof. □

Using the result in Lemma 3.4, we have the following result.

Lemma 3.5.

$$\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+\left(\sum_{i=1}^{n}\beta_i-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)}\le\alpha.$$

Proof. We prove the result in Lemma 3.5 by contradiction. Assume

$$\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+\left(\sum_{i=1}^{n}\beta_i-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)}>\alpha.$$

By statement (3) in Lemma B.1, we have

$$\begin{aligned}
&\frac{Q+\left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}B_{i,j}-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}F_i\left(\sum_{j=1}^{\hat{v}_i}B_{i,j}\right)}\\
=&\frac{Q+\left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\beta_{i,j}-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}+\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}(B_{i,j}-\beta_{i,j})p_{\max}}{\sum_{i=1}^{n}F_i\left(\sum_{j=1}^{\hat{v}_i}\beta_{i,j}\right)+\sum_{i=1}^{n}\left[F_i\left(\sum_{j=1}^{\hat{v}_i}B_{i,j}\right)-F_i\left(\sum_{j=1}^{\hat{v}_i}\beta_{i,j}\right)\right]}\\
\ge&\frac{Q+\left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\beta_{i,j}-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}+\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}(B_{i,j}-\beta_{i,j})p_{\max}}{\sum_{i=1}^{n}F_i\left(\sum_{j=1}^{\hat{v}_i}\beta_{i,j}\right)+\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}(B_{i,j}-\beta_{i,j})\frac{p_{\max}}{\alpha}}\\
>&\alpha.
\end{aligned}$$

During the lifetime of the j-th virtual storage of the i-th reservation period, the minimum electricity price is $\hat{\xi}_{i,j}$. When the procurement amount during the i-th reservation period is $\sum_{j=1}^{\hat{v}_i}B_{i,j}$, the cost of the optimal algorithm satisfies

$$\sum_{i=1}^{n}F_i\left(\sum_{j=1}^{\hat{v}_i}B_{i,j}\right)\ge\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{\xi}_{i,j}B_{i,j}.$$

Thus, we have

$$\begin{aligned}
&\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+\left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}B_{i,j}-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{\xi}_{i,j}B_{i,j}}\\
=&\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\left[\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+\left(B_{i,j}-\hat{b}_{i,j}\right)p_{\max}\right]}{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{\xi}_{i,j}B_{i,j}}>\alpha.
\end{aligned}$$

That means that there is at least a pair of i and j such that

$$\frac{\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+(B_{i,j}-\hat{b}_{i,j})p_{\max}}{\hat{\xi}_{i,j}B_{i,j}}>\alpha.$$

The above equation contradicts the results in Lemma 3.4. This completes the proof. □

Using the result in Lemma 3.5 and Equation (16), we have

$$\frac{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db+\left(D-\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum_{i=1}^{n}F_i(\beta_i)+\left(D-\sum_{i=1}^{n}\beta_i\right)\cdot\tilde{p}}\le\alpha.$$

Thus, we have

$$\texttt{cost(BatMan)}\le\alpha\cdot\texttt{cost(OPT)}+\hat{b}p_{\max}\le\alpha\cdot\texttt{cost(OPT)}+Bp_{\max},$$

where $Bp_{\max}$ is a constant. And the proof of Theorem 1.1 is complete.

3.2.3 The Optimality of the Competitive Ratio. By setting $d(t)=0, t\in\{1,\dots,T-1\}$ and $d(T)=B$, OLIM degenerates to the k-min search problem, whose optimal competitive ratio [30, Theorem 2] is exactly equal to that of `BatMan`. This directly results that α is a lower bound for the OLIM as a generalized k-min search problem.

3.3 `BatManRate`: An Online Algorithm with Rate Constraints

In this section, we design `BatManRate` that adds input and output rate constraints to `BatMan`. In Algorithm 2, the pseudocode of `BatManRate` is summarized. While the general flow is similar to that of `BatMan`, `BatManRate` has two major extensions:

First, `BatManRate` intelligently sets the capacity of virtual storages to respect output rate constraints. The high-level intuition to set the capacity of virtual storage is that output (discharge) rate constraint limits the capability of using the storage in each slot, and hence it may not be possible to fully satisfy the demand by discharging the storage, so creating a virtual storage with capacity equal to the demand does not make sense. This is done by calling sub-procedure `InitVS` in Line 6 of `BatManRate`, with details explained in §3.3.1.

Second, `BatManRate` intelligently sets the value of reservation prices to respect the input (charge) rate constraints. The high-level intuition is that the input rate constraint limits the amount of stored asset in each slot. Hence, if the price is very cheap, `BatMan` might propose to store some amount that is beyond the capability of

Algorithm 2 The BatManRate algorithm for each $t \in \mathcal{T}$

1: // Initialization: just at first slot
2: $B_1 \leftarrow B$; $v \leftarrow 1$; $\xi_1 \leftarrow p_{\max}/\alpha$
// The main procedure for t
// Initialize a new virtual storage
3: **if** $d(t) > 0$ **then**
4: $v \leftarrow v + 1$
5: $\xi_v \leftarrow p_{\max}/\alpha$
6: $B_v \leftarrow$ InitVS $(p(t), d(t), \rho_d, v, \{\xi_i, B_i\}_{i=1:v-1}, \varepsilon_1)$
7: **end if**
// Calculate the initial value for procurement amount
8: $\hat{x}(t) \leftarrow \sum_{i=1}^{v} \left[G_{B_i}(p(t)) - G_{B_i}(\xi_i)\right]^+$
9: $x(t) \leftarrow \hat{x}(t)$
10: $p \leftarrow p(t)$
// Check active output rate constraint
11: **if** $\hat{x}(t) < [d(t) - \min\{b(t-1), \rho_d\}]^+$ **then**
12: $x(t) \leftarrow [d(t) - \min\{b(t-1), \rho_d\}]^+$
13: **end if**
// Check active input rate constraint
14: **if** $\hat{x}(t) > \rho_c + d(t)$ **then**
15: $x(t) \leftarrow \rho_c + d(t)$
16: $p \leftarrow$ CalRP $(d(t), \rho_c, v, \{\xi_i, B_i\}_{i=1:v}, \varepsilon_2)$
17: **end if**
18: $b(t) \leftarrow b(t-1) + x(t) - d(t)$
19: $\xi_i \leftarrow \min\{p, \xi_i\}, \quad \forall 1 \le i \le v$
20: **if** $b(t) = 0$ **then** $v \leftarrow 1$ and $\xi_1 \leftarrow p_{\max}/\alpha$

Algorithm 3 InitVS $(p(t), d(t), \rho_d, v, \{\xi_i, B_i\}_{i=1:v-1}, \varepsilon_1)$

1: $B_v \leftarrow 0$, $B'_v \leftarrow d(t)$
2: **while** $|B'_v - B_v| > \varepsilon_1$ **do**
3: $B_v \leftarrow B'_v$
4: $\hat{x}(t) \leftarrow \sum_{i=1}^{v} \left[G_{B_i}(p(t)) - G_{B_i}(\xi_i)\right]^+$
5: $B'_v \leftarrow d(t) - [d(t) - \rho_d - \hat{x}(t)]^+$
6: **end while**
7: **Output**: B_v

storage to input. Hence, BatManRate updates the reservation price based on the capability to store, i.e., input rate constraint. This is done by calling the sub-procedure CalRP in Line 18 of BatManRate, with details presented in §3.3.2.

3.3.1 Initializing the New Virtual Storage. At t, the *preferred procurement amount*, denoted as $\hat{x}(t)$, the amount without considering output rate, should be calculated as

$$\hat{x}(t) = \sum_{i=1}^{v} \left[G_{B_i}(p(t)) - G_{B_i}(\xi_i)\right]^+, \tag{19}$$

where B_v is the new capacity of virtual storage. Different from the BatMan that sets the capacity to $d(t)$, BatManRate subtracts $[d(t) - \rho_d - \hat{x}(t)]^+$, i.e., the additional amount due to output rate constraint, from the capacity, hence

$$B_v = d(t) - [d(t) - \rho_d - \hat{x}(t)]^+. \tag{20}$$

Algorithm 4 CalRP $(d(t), \rho_c, v, \{\xi_i, B_i\}_{i=1:v}, \varepsilon_2)$

1: $p \leftarrow p_{\min}$, $p' \leftarrow p_{\max}/\alpha$;
2: **while** $|p' - p| > \varepsilon_2$ **do**
3: $z \leftarrow \sum_{i=1}^{v} \left[G_{B_i}((p+p')/2) - G_{B_i}(\xi_i)\right]^+$
4: **if** $z > \rho_c + d(t)$ **then** $p \leftarrow (p+p')/2$ **else** $p' \leftarrow (p+p')/2$
5: **end while**
6: **Output**: p

Equations (19) and (20) show that B_v and $\hat{x}(t)$ are dependent on each other. To address this, we devise the sub-procedure InitVS (Algorithm 3) to calculate the capacity of the new virtual storage. InitVS captures this dependency and updates determining B_v and $x(t)$ using a simple search algorithm in iterative manner with parameter ε_1 as the stopping criteria (ε_1 can be arbitrarily close to 0). In Appendix C, we prove that InitVS converges to a solution to Equations (19) and (20).

Lines 9-14 of BatManRate calculates the procurement amount $x(t)$ by taking into account rate constraints in the following cases:

(1) *Inactive output and input rate, Line* 9. In this case, $[d(t) - \min\{b(t-1), \rho_d\}]^+ \le \hat{x}(t) \le \rho_c + d(t)$, hence, $x(t) = \hat{x}(t)$.

(2) *Active output rate constraint, Line* 11. In this case, $\hat{x}(t)$ fails to satisfy the demand, so we set the actual procurement amount $x(t) = [d(t) - \min\{b(t-1), \rho_d\}]^+$.

(3) *Active input rate constraint, Line* 14. In this case, $\hat{x}(t)$ will be beyond the input rate of storage, hence $x(t) = \rho_c + d(t)$.

3.3.2 Calculating the Reservation Price. The final step is to update the reservation price ξ_i for each virtual storage. For cases (1) and (2), the reservation price for each virtual storage is updated similar to that of BatMan, i.e., $\min\{\xi_i, p(t)\}$ (Lines 10 and 19). For case (3), ξ_i is updated as follows. Let p be the updated reservation price, whose value is the solution to the following equation

$$\sum_{i=1}^{v_t} \left[G_{B_i}(p) - G_{B_i}(\xi_i)\right]^+ = \rho_c + d(t).$$

BatManRate solves the above equation for p by calling the sub-procedure CalRP (Algorithm 4) in iterative manner, with parameter ε_2 as the stopping criteria (ε_2 can be arbitrarily close to 0).

3.3.3 Competitive Analysis of BatManRate. Last but not the least, in this section, we briefly explain the proof sketch for the result in Theorem 1.2. The rigorous proof is given in Appendix D.

The proof sketch is as follows. We first show that in worst case, the output rate is not active. Then, we find an upper bound on the cost of BatManRate similar to that of in Lemma 3.2. This upper bound gets affected by input rate constraints. The rest is akin to the competitive analysis of BatMan.

4 EMPIRICAL EVALUATION

To illustrate the performance of our algorithms, we focus on the example of energy procurement, and evaluate their performance using various traces. Our results answers the following questions:

(1) How does the empirical cost ratio of BatMan *compare to the theoretical competitive ratio?* We find that BatMan achieves a significantly smaller average cost ratio than the worst-case competitive ratio guarantee provided by our theoretical analysis (Observation 1).

(2) How does BatMan *compare to the existing algorithms?* We find that BatMan outperforms all the baseline and existing algorithms [41, 43, 45] by 7%-15%, on average.

(3) How sensitive is BatMan *to various parameters such as the penetration of renewable?* Our experiments demonstrate that BatMan is lightly affected by these parameters as compared to substantial performance fluctuations in alternative algorithms (Observations 4).

(4) How does BatManRate *and compare to the basic* BatMan*?* We find that the empirical cost ratio of BatManRate is only slightly worst than BatMan once rate constraints are tight (Observation 6).

4.1 Experimental Methodology

We perform trace-based simulations with extensive data traces. We use Akamai traces for the energy demand, four different electricity market for the energy prices, and nearby renewable generations to evaluate the results with more uncertainty from renewable generation. The details of data traces are given in Appendix E.

Table 1: Summary of algorithms that are evaluated

Our proposed online algorithms	
BatMan	Basic online algorithm (§3.1)
BatManRate	Online algorithm with rate constraints (§3.3)
PreDay	A simple data-driven approach to use optimal solution for the previous day for the current day
Other algorithms for comparison	
OPT	Optimal offline solution with storage
NoSTR	Optimal offline solution without storage
OnFix [45]	State-of-the-art online algorithm with fixed threshold price
LypOpt [43]	State-of-the-art online Lyapunov-based algorithm

4.1.1 Comparison Algorithms. We implemented our algorithms and several other state-of-the-art algorithms for comparison as described below (see Table 1).

▷ (OPT) Optimal offline algorithm with storage by solving OLIM in §2. Since OPT represents the best achievable cost for the given inputs, all other algorithms are evaluated by computing *empirical cost ratio* which is the ratio of the cost of the algorithm with the cost of OPT. The cost ratio is always greater than equal to 1 and lower the cost ratio of an algorithm, the better the performance.

▷ (NoSTR) A baseline scheme that simply satisfies the net energy demand from the grid, assuming no storage is available. The cost ratio of NoSTR quantifies the maximum benefit of having storage.

▷ (PreDay) Our data-driven approach that uses the optimal derived for the previous day for the current day by projecting into a feasible range (satisfying capacity, demand, and rate constraints). PreDay is representative of a statistical approach that uses historical statistics to inform future decisions.

▷ (OnFix) Existing sub-optimal online algorithm [45] is a simple strategy that uses a fixed threshold of $p = \sqrt{p_{\max} \times p_{\min}}$ as the purchasing threshold and fully charges the storage if the current price is lower than p, otherwise, discharges the storage as much as possible and purchases the remaining amount from the grid.

▷ (LypOpt) Lyapunov-based approach [41, 43] that uses Lyapunov optimization to solve OLIM. Note that [43] considers load-balancing among multiple data centers as well, and to have a fair comparison, we focus on single data center model in [43], and with this reduction both algorithms in [41, 43] become similar.

Parameter Settings. Unless otherwise mentioned, we set the length of each slot to 5 minutes, according to FERC rule. The time horizon is 1 day, hence, $T = 12{\times}24 = 288$. We set the time horizon to 1 day to potentially see the impact of daily patterns in PreDay. The capacity of energy storage is set to $C = 18 \times \max_{t \in \mathcal{T}} d(t)$, sufficient to power the data center for 1.5 hours at max net demand. In experiments with renewable, the renewable penetration for solar and wind is 50%. Finally, each data point in figures and tables corresponds to the average results of 30 runs (days) over a month, each with the corresponding demand, renewable generation, and market prices.

4.2 BatMan vs. Alternative Algorithms

In Table 2, the empirical cost ratio of 5 algorithms (NoSTR, PreDay, LypOpt, OnFix, and BatMan) are reported across a broad set of settings: (i) four different locations; (ii) four different seasons; (iii) and with/without renewables. We report the notable observations:

Observation 1. BatMan *achieves a significantly smaller average cost ratio than the worst-case competitive ratio guarantee provided by our theoretical analysis.* The average theoretical competitive ratio (α in Equation (2)) for year-round experiments over four locations is 3.76, while the empirical cost ratios for BatMan are much smaller, i.e., 1.21 for no renewable and with solar, and 1.23 for wind.

Observation 2. BatMan *outperforms the alternative algorithms, when averaged across the entire year and all four locations, and with/without renewables and it is close to offline optimal* OPT. For example, in the case with wind as the renewable source, the last row of Table 2 shows that BatMan outperforms NoSTR by 15%, PreDay by 10%, LypOpt by 8%, and OnFix by 7.5%. Further, on average over the whole year BatMan achieves a cost ratio of 1.21 to 1.23, i.e., a cost that is within 21–23% of the cost of OPT. However. there are a few settings that other algorithms outperform BatMan, e.g., PreDay in Frankfurt/DE. The reason is investigated in the next observation.

Observation 3. PreDay *is the best algorithm in settings with low price uncertainty and recurring daily price patterns.* To elaborate this observation, we need to further investigate the dynamics of the real-time prices in DE market. Figure 1 shows the real-time prices in three days in August 2017 for NYISO with high fluctuation ratio and DE Market with low price fluctuations. Once can see that in DE Market the prices do not fluctuate a lot and there is almost a regular daily pattern. This regular daily pattern is the key to PreDay's good performance since it uses the previous day values to derive the procurement plan for today. However, the irregular pattern in NYISO and other markets (as shown in Figure 1) motivates our general BatMan approach, since relying on the past information or stochastic modeling is less effective in these real-world markets. Predictably, PreDay does not perform as well in CAISO and NYISO markets in Table 1 (the cost ratio of 1.63 and 1.46 for PreDay as compared to 1.35 and 1.31 for BatMan).

Observation 4. *With increased uncertainty due to renewable penetration, the performance of* BatMan *is robust, however, the performance of* PreDay *degrades substantially.* The performance of BatMan slightly degrades with injection of 50% renewable. The performance of PreDay, however, degrades substantially (e.g., from 1.31 to 1.37 for wind). To further elaborate this, in Figure 4, we compare the performance of BatMan and PreDay in different seasons

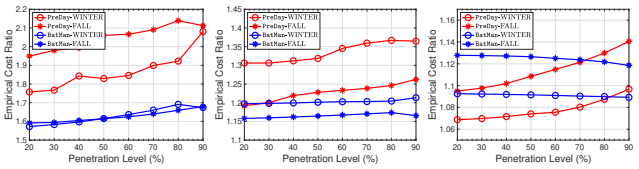
Table 2: The empirical cost ratio of different algorithms in different markets and different seasons

	City/Market	θ	α, Eq. (2)	Cost ratio for no renewables					Cost ratio for 50% wind penetration					Cost ratio for 50% solar penetration				
				NoSTR	PreDay	LypOpt	OnFix	BatMan	NoSTR	PreDay	LypOpt	OnFix	BatMan	NoSTR	PreDay	LypOpt	OnFix	BatMan
Winter	**Los Angles/CAISO**	110.00	7.74	1.88	1.60	1.79	1.53	1.44	2.06	2.13	1.74	1.68	1.51	1.93	1.67	1.71	1.56	1.44
	New York/NYISO	26.89	3.99	1.52	1.46	1.47	1.45	1.29	1.60	1.54	1.49	1.55	1.33	1.55	1.49	1.42	1.48	1.29
	Dallas/ERCOT	15.83	3.13	1.23	1.15	1.13	1.13	1.15	1.26	1.16	1.15	1.16	1.13	1.26	1.17	1.16	1.15	1.13
	Frankfurt/DE	2.22	1.36	1.11	1.03	1.10	1.09	1.07	1.13	1.04	1.12	1.10	1.08	1.11	1.03	1.10	1.09	1.07
	Average	**38.73**	**4.05**	**1.43**	**1.31**	**1.37**	**1.29**	**1.24**	**1.51**	**1.47**	**1.37**	**1.36**	**1.26**	**1.46**	**1.34**	**1.35**	**1.31**	**1.23**
Spring	**Los Angles/CAISO**	96.95	7.29	1.99	1.87	1.51	1.54	1.34	2.05	1.73	1.68	1.63	1.39	2.01	1.94	1.82	1.56	1.35
	New York/NYISO	28.79	4.11	1.54	1.42	1.47	1.44	1.33	1.58	1.45	1.49	1.49	1.34	1.57	1.45	1.52	1.49	1.34
	Dallas/ERCOT	10.09	2.56	1.27	1.17	1.22	1.15	1.15	1.33	1.21	1.27	1.19	1.17	1.29	1.16	1.20	1.17	1.14
	Frankfurt/DE	2.04	1.31	1.07	1.03	1.07	1.07	1.05	1.08	1.03	1.07	1.08	1.05	1.08	1.03	1.07	1.08	1.05
	Average	**34.47**	**3.81**	**1.47**	**1.37**	**1.32**	**1.30**	**1.22**	**1.51**	**1.35**	**1.38**	**1.34**	**1.24**	**1.49**	**1.39**	**1.40**	**1.32**	**1.22**
Summer	**Los Angles/CAISO**	25.10	3.86	1.56	1.36	1.51	1.41	1.32	1.56	1.37	1.51	1.42	1.33	1.60	1.39	1.54	1.44	1.34
	New York/NYISO	19.96	3.48	1.33	1.26	1.31	1.34	1.24	1.35	1.29	1.28	1.38	1.26	1.35	1.27	1.30	1.38	1.24
	Dallas/ERCOT	5.91	2.03	1.17	1.07	1.14	1.12	1.09	1.20	1.10	1.14	1.13	1.09	1.18	1.07	1.14	1.13	1.07
	Frankfurt/DE	2.31	1.37	1.08	1.03	1.08	1.07	1.06	1.09	1.04	1.09	1.08	1.06	1.09	1.03	1.08	1.08	1.07
	Average	**13.32**	**2.68**	**1.29**	**1.18**	**1.26**	**1.23**	**1.18**	**1.30**	**1.20**	**1.25**	**1.25**	**1.19**	**1.30**	**1.19**	**1.27**	**1.25**	**1.18**
Fall	**Los Angles/CAISO**	51.84	5.42	1.58	1.70	1.39	1.42	1.29	1.64	1.89	1.43	1.47	1.31	1.63	1.80	1.44	1.45	1.30
	New York/NYISO	36.04	4.57	1.71	1.71	1.67	1.61	1.40	1.81	1.77	1.75	1.75	1.43	1.74	1.72	1.64	1.65	1.40
	Dallas/ERCOT	7.26	2.22	1.22	1.12	1.20	1.13	1.08	1.23	1.15	1.16	1.14	1.09	1.23	1.12	1.16	1.13	1.08
	Frankfurt/DE	2.12	1.33	1.12	1.05	1.11	1.09	1.09	1.15	1.08	1.12	1.11	1.10	1.12	1.06	1.12	1.09	1.09
	Average	**24.31**	**3.38**	**1.41**	**1.40**	**1.34**	**1.31**	**1.21**	**1.46**	**1.47**	**1.36**	**1.36**	**1.23**	**1.43**	**1.43**	**1.34**	**1.33**	**1.22**
Year	**Los Angles/CAISO**	70.97	6.28	1.75	1.63	1.55	1.47	1.35	1.83	1.78	1.59	1.55	1.39	1.79	1.70	1.63	1.50	1.36
	New York/NYISO	27.92	4.06	1.52	1.46	1.48	1.46	1.31	1.58	1.51	1.50	1.54	1.34	1.55	1.48	1.47	1.50	1.32
	Dallas/ERCOT	9.77	2.53	1.22	1.13	1.17	1.13	1.12	1.25	1.15	1.18	1.16	1.12	1.24	1.13	1.17	1.14	1.11
	Frankfurt/DE	2.17	1.34	1.09	1.04	1.09	1.08	1.07	1.11	1.05	1.10	1.09	1.07	1.10	1.04	1.09	1.08	1.07
	Average	**27.71**	**3.76**	**1.40**	**1.31**	**1.32**	**1.28**	**1.21**	**1.44**	**1.37**	**1.34**	**1.33**	**1.23**	**1.42**	**1.34**	**1.34**	**1.30**	**1.21**

Table 3: Comparison of different algorithms using different energy storage technologies

City/Market	θ	Lithium-Ion ($\rho_c/B = 0.35$)					Lead-Acid ($\rho_c/B = 0.2$)					Compressed Air Energy Storage ($\rho_c/B = 0.05$)				
		NoSTR	PreDay	LypOpt	OnFix	BatManRate	NoSTR	PreDay	LypOpt	OnFix	BatManRate	NoSTR	PreDay	LypOpt	OnFix	BatManRate
Los Angles/CAISO	25.10	1.49	1.34	1.44	1.33	1.31	1.47	1.33	1.44	1.32	1.32	1.43	1.31	1.42	1.29	1.32
New York/NYISO	19.96	1.27	1.21	1.24	1.24	1.21	1.25	1.19	1.22	1.22	1.20	1.23	1.17	1.20	1.19	1.18
Dallas/ERCOT	7.26	1.14	1.10	1.11	1.13	1.07	1.14	1.10	1.13	1.13	1.07	1.13	1.09	1.13	1.12	1.07
Frankfurt/DE	2.12	1.07	1.03	1.05	1.08	1.05	1.06	1.04	1.05	1.08	1.05	1.06	1.03	1.05	1.08	1.05
Average	**13.32**	**1.24**	**1.17**	**1.21**	**1.19**	**1.16**	**1.23**	**1.16**	**1.21**	**1.18**	**1.16**	**1.21**	**1.15**	**1.20**	**1.17**	**1.15**

and locations as the penetration level varies. The result signifies the robust performance of BatMan and degradation of PreDay with increased renewable penetration.



(a) New York (b) Dallas (c) Frankfurt

Figure 4: The impact of renewable penetration

Observation 5. *The seasonal and locational patterns result in different degrees of uncertainty, thereby impact the performance of the algorithms substantially and increased uncertainty increases the cost ratio of algorithms.* For example, in BatMan, the year-round cost ratio in Los Angles as the most uncertain location (highest $\theta = 70.97$, on average) is 1.47, while the same value for Frankfurt (lowest $\theta = 2.17$) as the least uncertain one is 1.07. As for seasonal variations, BatMan achieves the cost ratio of 1.18 in the summer as the least uncertain season (with average $\theta = 13.32$), while this value is1.24 in the winter as the most uncertain scenario (with $\theta = 38.73$).

Evaluation of BatManRate. To evaluate the performance of the second algorithm BatManRate, we consider identical charging and discharge rates, i.e., $\rho_c = \rho_d$, and normalize it against the storage capacity. Hence, we define $\rho = \rho_c/B$ as a measure of the rate at which an energy storage is charged/discharged relative to its capacity. A broad spectrum of storage technologies are integrated in data centers, each with a different ρ. To obtain practical values, we use the energy density as the normalized capacity, and power density as the normalized discharge rate from [50]. We choose four common categories based on their ρ values [51, 52]: (1) Compressed Air Energy Storage with $\rho \approx 0.05$; (2) Lead-Acid with $\rho \approx 0.2$; (3) Lithium-Ion with $\rho \approx 0.35$, and (4) Flywheels $\rho \approx 1$. For Flywheels, there is no rate constraints and it reduces to BatMan. Hence, we investigate the performance of BatManRate for the first three technologies, and report the results in Table 3.

Observation 6. *The performance of* BatManRate *improves as ρ increases, i.e., the rate constraints becomes more relax, while the performance of* PreDay *exhibits no regular pattern.* This observation is inferred from the results in Table 3 that reports the results for three representative energy storage technologies.

5 CONCLUSION

We developed two competitive algorithms for online linear programming with inventory management constraints. We proved that both algorithms achieve the best possible competitive ratio. We evaluated the proposed algorithms using extensive data-traces from the application of energy procurement in data centers. As future work, we plan to extend the results and tackle maximization version of the problem, and general convex cost function.

REFERENCES

[1] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis.* Cambridge University Press, 1998.

[2] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[3] L. Yang, L. Deng, M. H. Hajiesmaili, C. Tan, and W. S. Wong, "An optimal algorithm for online non-convex learning," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, p. 25, 2018.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[5] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proceedings of the fourth international conference on Future energy systems*, pp. 137–148, 2013.

[6] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2013.

[7] S. Albers, "On energy conservation in data centers," in *Proc. of ACM SPAA*, pp. 35–44, 2017.

[8] X. Ren, P. London, J. Ziani, and A. Wierman, "Datum: Managing data purchasing and data placement in a geo-distributed data market," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 893–905, 2018.

[9] S. Albers and J. Quedenfeld, "Optimal algorithms for right-sizing data centers," in *Proc. of ACM SPAA*, pp. 363–372, 2018.

[10] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-aware online economic dispatching for microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 323–335, 2018.

[11] R. Zhou, Z. Li, and C. Wu, "An online procurement auction for power demand response in storage-assisted smart grids," in *Proc. of IEEE INFOCOM*, pp. 2641–2649, 2015.

[12] K. Khezeli and E. Bitar, "Risk-sensitive learning and pricing for demand response," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6000–6007, 2018.

[13] M. H. Hajiesmaili, M. Chen, E. Mallada, and C.-K. Chau, "Crowd-sourced storage-assisted demand response in microgrids," in *Proc. of ACM eEnergy*, pp. 91–100, 2017.

[14] Z. Mao, C. E. Koksal, and N. B. Shroff, "Optimal online scheduling with arbitrary hard deadlines in multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 177–189, 2016.

[15] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, p. 23, 2017.

[16] K. Kobayashi, S. Miyazaki, and Y. Okabe, "A tight bound on online buffer management for two-port shared-memory switches," in *Proc. of ACM SPAA*, pp. 358–364, 2007.

[17] B. Feldkord and F. Meyer auf der Heide, "Online facility location with mobile facilities," in *Proc. of ACM SPAA*, pp. 373–381, 2018.

[18] J. W.-T. Chan, F. Y. Chin, D. Ye, and Y. Zhang, "Online frequency allocation in cellular networks," in *Proc. of ACM SPAA*, pp. 241–249, 2007.

[19] M. H. Hajiesmaili, L. Deng, M. Chen, and Z. Li, "Incentivizing device-to-device load balancing for cellular networks: An online auction design," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 265–279, 2017.

[20] G. Even, M. Medina, and D. Rawitz, "Online generalized caching with varying weights and costs," in *Proc. of ACM SPAA*, pp. 205–212, 2018.

[21] "Google data center in Changhua County, Taiwan." available at https://www.google.com/about/datacenters/inside/locations/changhua-county/.

[22] "Tesla's Powerpack proposes battery power for data centers." available at https://www.datacenterdynamics.com/analysis/teslas-powerpack-proposes-battery-power-for-data-centers/.

[23] "Google data center in St. Ghislain, Belgium." available at https://www.google.com/about/datacenters/inside/locations/st-ghislain/.

[24] "Environmental responsibility report - apple." available at https://www.apple.com/environment/pdf/Apple_Environmental_Responsibility_Report_2018.pdf.

[25] M. Ghamkhari, A. Wierman, and H. Mohsenian-Rad, "Energy portfolio optimization of data centers," *IEEE Trans. Smart Grid*, 2016.

[26] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *Proc. of ACM eEnergy*, pp. 51–61, 2014.

[27] E. Mohr, I. Ahmad, and G. Schmidt, "Online algorithms for conversion problems: a survey," *Surveys in Operations Research and Management Science*, vol. 19, no. 2, pp. 87–104, 2014.

[28] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal search and one-way trading online algorithms," *Algorithmica*, vol. 30, no. 1, pp. 101–139, 2001.

[29] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg, "Online auctions and generalized secretary problems," *ACM SIGecom Exchanges*, vol. 7, no. 2, p. 7, 2008.

[30] J. Lorenz, K. Panagiotou, and A. Steger, "Optimal algorithms for k-search with application in option pricing," *Algorithmica*, vol. 55, no. 2, pp. 311–328, 2009.

[31] N. Buchbinder and J. Naor, "Online primal-dual algorithms for covering and packing," *Mathematics of Operations Research*, vol. 34, no. 2, pp. 270–286, 2009.

[32] "CAISO electricity market." available at https://www.caiso.com/.

[33] "NYISO electricity market." available at http://www.nyiso.com.

[34] "ERCOT electricity market." available at http://www.ercot.com.

[35] "German electricity market." available at https://www.smard.de/en/.

[36] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: a platform for high-performance internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[37] A. Dobos, *PVWatts version 5 manual.* National Renewable Energy Laboratory Golden, CO, 2014.

[38] "Eastern and western data sets." available at https://www.nrel.gov/grid/eastern-western-wind-data.html.

[39] "Open power system data." available at https://data.open-power-system-data.org/.

[40] "Wholesale electricity market design initiatives in the united states: Survey and research needs," *EPRI,Technical Results, available at https:// www.epri.com/#/pages/product/000000003002009273/*, 2016.

[41] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proc. ACM SIGMETRICS*, 2011.

[42] L. Huang, J. Walrand, and K. Ramchandran, "Optimal demand response with energy storage management," in *Proc. IEEE SmartGridComm*, pp. 61–66, 2012.

[43] Y. Guo and Y. Fang, "Electricity cost saving strategy in data centers by using energy storage," *IEEE Transactions Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1149–1160, 2013.

[44] P. M. van de Ven, N. Hegde, L. Massoulié, and T. Salonidis, "Optimal control of end-user energy storage," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 789–797, 2013.

[45] C.-K. Chau, G. Zhang, and M. Chen, "Cost minimizing online algorithms for energy storage management with worst-case guarantee," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2691–2702, 2016.

[46] R. Evans and J. Gao, "DeepMind AI reduces google data centre cooling bill by 40%," *https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/*, 2016.

[47] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, pp. 175–186, 2012.

[48] L. Yang, M. H. Hajiesmaili, H. Yi, and M. Chen, "Hour-ahead offering strategies in electricity market for power producers with storage and intermittent supply," in *Proc. ACM SIGMETRICS*, pp. 21–22, 2017.

[49] L. Yang, W. S. Wong, and M. H. Hajiesmaili, "An optimal randomized online algorithm for qos buffer management," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, p. 36, 2017.

[50] "Comparison of commercial battery types." available at https://en.wikipedia.org/wiki/Comparison_of_commercial_battery_types.

[51] S. Chalise, A. Golshani, S. R. Awasthi, S. Ma, B. R. Shrestha, L. Bajracharya, W. Sun, and R. Tonkoski, "Data center energy systems: Current technology and future direction," in *Proc. of IEEE PES*, 2015.

[52] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy, "Energy storage in datacenters: what, where, and how much?," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, pp. 187–198, 2012.

[53] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, no. 12, pp. 33–37, 2007.

[54] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar, "Using batteries to reduce the power costs of internet-scale distributed networks," in *Proc. of ACM SoCC*, 2012.

A PROOF OF THEOREM 3.1

To proof, we show that `BatMan` respects all the constraints in OLIM. First, it respects the demand covering constraint, i.e., $x(t) \geq d(t) - b(t-1)$, by the projection in Equation (13). Second, we show that `BatMan` always respects the capacity constraints. At time slot t which lies in a reservation period (see Definition 3), the total amount of purchased asset from the beginning of current reservation period is equal to $\sum_{i=1}^{v} G_{B_i}(\xi_i)$, which is less than or equal to $\sum_{i=1}^{v} B_i$. The demand from the beginning of current reservation period is $\sum_{i=2}^{v} B_i$. The asset stored in the physical storage is $\sum_{i=1}^{v} G_{B_i}(\xi_i) - \sum_{i=2}^{v} B_i$, which is less than or equal to $\sum_{i=1}^{v} B_i - \sum_{i=2}^{v} B_i = B_1$ according to the definition of $G_{B_i}(\xi_i)$. B_1 is the capacity of the physical storage. That is, the amount of reserved power always respects the capacity constraint.

B PROOFS RELATED TO ANALYSIS OF BATMAN

B.1 Proof of Lemma 3.2

For each virtual storage, BatMan stores asset only if the market price is less than the reservation price. Hence, the cost of the stored assets in j-th storage is less than $\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db$. By aggregation over n reservation periods and virtual storages, we can compute the aggregate cost incurred by BatMan as

$$\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db, \tag{21}$$

where there are $\hat{v}_i$ virtual storage units at reservation period i, $\hat{\xi}_{i,j}$ and $\hat{b}_{i,j}$ is the final storage level of virtual storage j at reservation period i. The additional amount of electricity needed to satisfy the demand in the idle period is equal to $D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j} + \hat{b}$, where D be the total demand during the time horizon, i.e., $D = \sum_{t\in\mathcal{T}} d(t)$, and $\hat{b}$ be the final storage level of the physical storage, i.e., $\hat{b} = b(T)$. Hence, the cost of the online algorithm during the idle period is at most

$$\left(D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j} + \hat{b}\right)p_{\max}, \tag{22}$$

Adding (21) and (22) completes the proof.

B.2 Proof of Lemma 3.3

Similar to the online algorithm, the cost of an optimal offline solution, denoted as cost(OPT), can be also split into two parts. To characterize a lower bound for the offline optimum, let us define $F_i(\beta)$ as the minimum cost of purchasing β units of asset during the i-th reservation period.

Let β_i be the asset purchased by the optimal offline solution during the i-th reservation period. Obviously, the cost of the optimal offline solution during the i-th reservation period is at least $F_i(\beta_i)$. Let $\tilde{p}$ be the minimum price during the idle periods, then we have that the cost of the offline algorithm is lower bounded by $\sum_{i=1}^{n} F_i(\beta_i) + (D - \sum_{i=1}^{n}\beta_i)\times\tilde{p}$. This proof is complete.

B.3 Additional Results Required to Prove Lemma 3.5

We state the following lemma on the properties of function $G^{-1}_{B_i}(b)$ and $F_i(\beta)$ to facilitate the proof of Lemma 3.5. Let $B_{i,j}$ be the capacity of the j-th storage and $\hat{b}_{i,j}$ be the final storage level at reservation period i.

Lemma B.1. *Defining $\hat{b}_i$ as the initial state of the storage of the offline algorithm at i-th reservation period, there is a worst-case input instance such that for all $0 < \beta < \beta' < \sum_{j=1}^{\hat{v}_i} B_{i,j} - \hat{b}_i$, we have*

(1) $F_i(0) = 0$;
(2) $F_i(\beta') > F_i(\beta)$;
(3) $F_i(\beta') - F_i(\beta) \le \frac{p_{\max}}{\alpha}(\beta' - \beta)$.

Proof. Statements (1) and (2) are straightforward.

Assume there is a worst instance $\omega = [\langle p(t), d(t)\rangle]_{t\in\mathcal{T}}$. The adversary can construct a new instance ω' (as shown in Equation (23)) by adding one time slot before each time slot of instance ω. Note that this is possible since the adversary can set the length of time horizon. The market prices for the newly added time slots is $p_{\max}/\alpha$ and the demand is always equal to zero.

In this way, we construct a new instance under which the cost of the online algorithm does not change, and that of the offline optimal solution will not increase. Thus, ω' is also the worst instance. Let $F_i(\beta)$ be the minimum cost when buying β units of asset during the i-th reservation period. When the optimal policy buys another $\beta' - \beta$, $\beta' < \sum_{j=1}^{\hat{v}_i} B_{i,j} - \hat{b}_i$, units of asset, the cost will not be larger than $\frac{p_{\max}}{\alpha}(\beta' - \beta)$, since the optimal policy can buy asset at any newly added time slots. In this way we prove that, there is a worst instance, such that $F_i(\beta') - F_i(\beta) \le p_{\max}/\alpha(\beta' - \beta)$, for $0 < \beta < \beta' < \sum_j \hat{b}_{i,j} - \hat{b}_i$. This completes the proof. □

C CONVERGENCE OF INITVS

Theorem C.1. *Given a market price $p(t) \in [p_{\min}, p_{\max}]$, InitVS converges to a feasible solution B_v and $\hat{x}(t)$ which satisfy Equations (19) and (20) simultaneously.*

Proof. It is easy to see that B'_v is always larger than or equal to B_v. And if the the value of $B'_v - B_v$ is larger than ε_1, the value of B_v will increase by at least ε_1. Thus, there must be an iteration such that $B'_v - B_v \le \varepsilon_1$. □

D COMPETITIVE ANALYSIS OF BATMANRATE

If $D = 0$, BatManRate can be easily proved α-competitive. Thus, we focus our analysis on the case $D > 0$.

Similar to the analysis for BatMan, we would like to upper bound the cost of BatManRate. To achieve this, first we give the following two lemmas which characterize properties of the worst instance for BatManRate. Lemma D.1 implies that in worst case, the output constraint is not active. Lemma D.2 characterizes an upper bound on the cost of BatManRate.

Lemma D.1. *Under the worst case, $\tilde{x}(t) = 0$, for $\forall t \in \mathcal{T}$, where $\tilde{x}(t) = [d(t) - \rho_d - \hat{x}(t)]^+$.*

Proof. We prove this lemma by contradiction. Assume there is a worst instance $\omega = [\langle p(t), d(t)\rangle]_{t\in\mathcal{T}}$, where $\tilde{x}(t) > 0$ for time slot t. We can construct a new instance which is the same as ω except at the t-th time slot. For time slot t, the demand is set to $x(t) - \delta$, where $\delta < \tilde{x}(t)$, and the market price is equal to $p(t)$. In this way, the cost of BatMan at time slot t will decrease by $p(t)\delta$. The costs on other time slots are unchanged, because the modification on the demand does not influence the capacity and reservation price of virtual storage according to the rules of BatManRate. On the other hand, the cost of OPT at time slot t will decrease by at least $p(t)\delta$, since the procurement amount of OPT is larger than δ. In this case, we have a new instance ω' under which the cost ratio is larger than that of the worst instance, contradicting the assumption. This completes the proof. □

Lemma D.1 implies that under the worst case the output constraint is not active. Then, we take into account the influence of the input rate constraint. Recall that in the basic version, BatMan, the procurement amount is always larger than or equal to $\hat{x}(t)$, which is computed in Equation (19). With input constraint, $\hat{x}(t)$ may not be satisfied, and the maximum procurement amount is limited by

$$\omega' = \left[\left\langle \frac{p_{\max}}{\alpha}, 0\right\rangle, \langle p(1), d(1)\rangle, \left\langle \frac{p_{\max}}{\alpha}, 0\right\rangle, \langle p(2), d(2)\rangle, \ldots, \left\langle \frac{p_{\max}}{\alpha}, 0\right\rangle, \langle p(t), d(t)\rangle, \left\langle \frac{p_{\max}}{\alpha}, 0\right\rangle\right]. \quad (23)$$

$\rho_c + d(t)$. We define $\mathcal{T}_r \subset \mathcal{T}$ be the set of time slots at which the input rate truncates the procurement amount. That is, the following equation holds for $t \in \mathcal{T}_r$.

$$\sum_{i \le v} \left[G_{B_i}(p(t)) - G_{B_i}(\xi_i)\right]^+ > \rho_c + d(t).$$

For $t \in \mathcal{T}_r$, we define $p'(t)$ as the value which satisfies the following equation.

$$\sum_{i \le v} \left[G_{B_i}(p'(t)) - G_{B_i}(\xi_i)\right]^+ = \rho_c + d(t), \text{ for } \forall t \in \mathcal{T}.$$

$p'(t)$ is the actual reservation price computed in Algorithm 4, and obviously, $p'(t) > p(t)$. Let $\mu(t) = p'(t) - p(t)$ denotes the difference between $p'(t)$ and $p(t)$.

Denote $x(t)$ and $x^*(t)$ as the amount of reserved asset by the online algorithm and the offline algorithm at time slot t, respectively. Similar to Lemma 3.2, we have the following lemma which upper bounds the cost of the BatManRate.

Lemma D.2. *The cost of* BatManRate *is upper bounded by*

$$cost(\texttt{BatManRate}) \le Q + \left(D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \hat{b}_{i,j} + \hat{b}\right)\cdot p_{\max} - \sum_{t\in\mathcal{T}_r} \mu(t)x(t),$$

where $Q = \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \int_0^{\hat{b}_{i,j}} G^{-1}_{B_{i,j}}(b)db.$

Proof. By Lemma D.1, we have that, under the worst case, $\tilde{x}(t) = 0$ and the amount of reserved asset is always less than or equal to the value computed in Equation (19). Based on the analysis in 3.2, we have that the cost of BatManRate is upper bounded by $Q + \left(D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \hat{b}_{i,j} + \hat{b}\right)\cdot p_{\max}$. Moreover, $\forall t \in \mathcal{T}_r$, the actual price is less than the reservation price by $\mu(t)$, so the above upper bound is further modified to $Q + \left(D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \hat{b}_{i,j} + \hat{b}\right)\cdot p_{\max} - \sum_{t\in\mathcal{T}_r} \mu(t)x(t)$. This completes the proof. □

With the above two lemmas, the competitive ratio of BatManRate is upper bounded by

$$\begin{aligned}
&\frac{\text{cost(BatManRate)} - \hat{b}p_{\max}}{\text{cost(OPT)}} \\
&\le \frac{Q + \left(D - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \hat{b}_{i,j}\right)\cdot p_{\max} - \sum_{t\in\mathcal{T}_r}\mu(t)x(t)}{\sum_{i=1}^{n} F'_i(\beta_i) + \left(D - \sum_{i=1}^{n}\beta_i\right)\cdot \tilde{p}} \\
&\le \max\left\{\frac{Q + \left(\sum_{i=1}^{n}\beta_i - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)\cdot p_{\max} - \sum_{t\in\mathcal{T}_r}\mu(t)x(t)}{\sum_{i=1}^{n} F'_i(\beta_i)}, \alpha\right\} \\
&\le \max\left\{\frac{Q + \left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} B_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max} - \sum_{t\in\mathcal{T}_r}\mu(t)x(t)}{\sum_{i=1}^{n} F'_i\left(\sum_{j=1}^{\hat{v}_i} B_{i,j}\right)}, \alpha\right\},
\end{aligned}$$

where $F'_i(\beta)$ is defined as the minimum cost of purchasing β units of asset during the i-th reservation period. The definition of $F'_i(\beta)$ is similar to that of $F_i(\beta)$ for for the basic version of the problem and it also respects the properties listed in Lemma B.1.

During the lifetime of the j-th virtual storage of the i-th reservation period, the minimum reservation price is $\hat{\xi}_{i,j}$. The cost of the optimal algorithm satisfies

$$\sum_{i=1}^{n} F'_i\left(\sum_{j=1}^{\hat{v}_i} B_{i,j}\right) \ge \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} \hat{\xi}_{i,j}B_{i,j} - \sum_{t\in\mathcal{T}_r}\mu(t)x^*(t).$$

Then, we have

$$\begin{aligned}
&\text{cr(BatManRate)} \\
&\le \max\left\{\frac{Q + \left(\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i} B_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max} - \sum_{t\in\mathcal{T}_r}\mu(t)x(t)}{\sum_{i=1}^{n}\sum_{j=1}^{\hat{v}_i}\hat{\xi}_{i,j}B_{i,j} - \sum_{t\in\mathcal{T}_r}\mu_t x^*(t)}, \alpha\right\}
\end{aligned}$$

The following lemma characterizes a bound on $x(t)/x^*(t)$.

Lemma D.3. *Under the worst case, we have that* $x(t)/x^*(t)$ *is less than or equal to the competitive ratio, for any* $t \in \mathcal{T}_r$.

Proof. Let $\omega = [\langle p(t), d(t)\rangle]_{t\in\mathcal{T}}$ be the worst instance and at time slot t, there is $x(t)/x^*(t) >$ cr(BatManRate). We can construct a new instance ω' by increasing the market price at time slot t by δ, where $\delta \le \mu(t)$. That is

$$\omega' = [\langle p(1), d(1)\rangle, \ldots, \langle p(t) + \delta, d(t)\rangle, \ldots, \langle p(T), d(T)\rangle].$$

Under instance ω', the cost of BatMan will increase by $x(t)\delta$, and that of OPT increase by less than $\frac{x(t)\delta}{\text{cr(BatManRate)}}$. In this way, we can

get a worse instance ω' than ω, contradicting the assumption that ω is the worst instance. This completes the proof. □

By the above lemma, we have that,

$$\frac{\sum\limits_{t\in\mathcal{T}_r}\tilde{p}_t x(t)}{\sum\limits_{t\in\mathcal{T}_r}\tilde{p}_t x^*(t)} \leq \mathsf{cr}(\mathtt{BatManRate}).$$

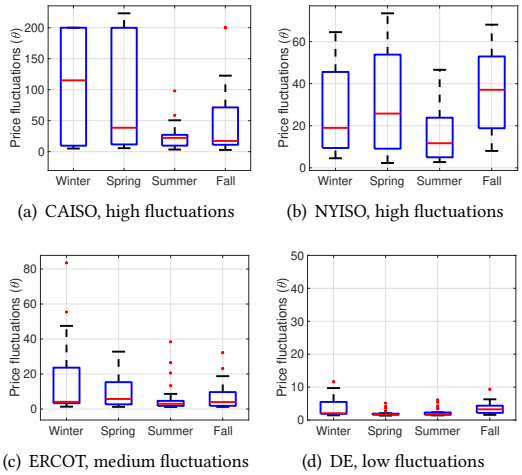
Then, there is,

$$\mathsf{cr}(\mathtt{BatManRate}) \leq \max\left\{\frac{Q+\left(\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{\hat{v}_i}B_{i,j}-\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{\hat{v}_i}\hat{b}_{i,j}\right)p_{\max}}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{\hat{v}_i}\hat{\xi}_{i,j}B_{i,j}},\alpha\right\}.$$

Combining with Lemma 3.4, we have $\mathsf{cr}(\mathtt{BatManRate}) \leq \alpha$. In this way, we prove that BatManRate is also α-competitive.

E ENERGY DEMAND, PRICE, AND RENEWABLE DATA TRACES

Data Center Energy Demand. We use a repository of demand traces from Akamai's server clusters in several data centers collected during a 31 day period from multiple locations around the world. The data includes the server load information from 973 data centers in 102 countries, collected every 5 minutes. To calculate energy consumption as a function of load, we use the standard linear model [53]. Let d_{idle} and d_{peak} be the energy consumption by an idle and a fully utilized server, respectively. Then, the energy (in kWh) consumed by a server serving normalized load $l \in [0,1]$ is $d(l) = d_{\text{idle}} + (d_{\text{peak}} - d_{\text{idle}}) \times l$. In our experiments, we use d_{idle} = 100kWh, and d_{peak} = 250kWh, representing energy proportionality factor, i.e., defined as $(d_{\text{peak}} - d_{\text{idle}})/d_{\text{peak}}$, of 0.6 [54]. We report the results of different algorithms for a selection of data centers in the four different cities: Los Angeles, New York, Dallas, Frankfurt. A representative 7-days snapshot of the energy consumption is depicted in Figure 2.



(a) CAISO, high fluctuations (b) NYISO, high fluctuations

(c) ERCOT, medium fluctuations (d) DE, low fluctuations

Figure 5: Price fluctuations in different seasons/locations

Energy Prices. We use the electricity prices from a local electricity market for each data center location, i.e., CAISO [32] for Los Angeles, NYISO [33] for New York, ERCOT [34] for Dallas, and German Electricity Market (abbreviated as DE in results) for Frankfurt. Note that FERC is forcing the U.S. electricity markets to transition to real-time markets with 5-minutes settlement intervals [40]. Currently CAISO and NYISO adapt this policy, and the rest are in the middle of this transition. To have a common settlement interval for all different markets, we set the length of each slot to 5 minutes, and for those that the current real-time market comes with different length (ERCOT with 15 minutes and DE with 1 hour intervals), we down-sample the market price readings to 5 minutes.

Recall that the performance of our algorithm is a function of parameter θ (see Equation (2)) as the price fluctuation ratio. Note that different markets exhibit different price fluctuations in different seasons. In Figure 5, the box plots for different markets in different seasons are shown. The results show that the fluctuations in spot prices in CAISO (Figure 5(a)) and NYISO (Figure 5(b)) are high, in ERCOT it is medium (Figure 5(c)), and in DE it is low (Figure 5(d)). Consequently, to have a comprehensive experimental study in different fluctuation patterns, we compare the performance of different algorithms in different markets and different seasons.

Renewable Data Traces. We evaluate the results of different algorithms in three different scenarios: (i) without any on-site renewable supply, (ii) with 50% penetration on-site wind generation; and (iii) with 50% penetration on-site solar generation. Note that with local renewable supply, the net energy demand, i.e., the total demand subtracted by the local renewable supply, must be procured from the grid with real-time pricing. Since the renewable supply is uncertain, the net demand in cases (ii) and (iii) will be more uncertain (as depicted in Figure 2(b)).

We use the solar data from PVWatts [37] and obtain the hourly solar radiation in different seasons. We match each data center with solar readings from a location as close to it as possible. The exact distance from data center to the location from where the readings were obtained is show in Table 4. We scale the values such that 50% of the total demand is satisfied by solar panels. We set the parameters according to the default values [37, Table 2]. While the spot prices and energy demand readings are 5-minutes, the solar data is hourly. Hence, we make an assumption that the solar data is almost constant during each hour and use the hourly values for each 5 minute slots. The wind traces for the U.S. locations are obtained from Eastern and Western data sets [38], and for European location are obtained from Open Power System Data [39]. A summary of locations, markets, and distances to renewables is listed in Table 4.

Table 4: Summary of data center locations, markets, and nearby solar and wind power plants used in experiments

City	Market	Dist. from solar	Dist. from wind
Los Angeles	CAISO	80 mi.	48 mi.
New York	NYISO	37 mi.	52 mi.
Dallas	ERCOT	63 mi.	145 mi.
Frankfurt	DE	35 mi.	-